FOCS Homework for Day 9

1. For each of the following languages, decide whether it is regular

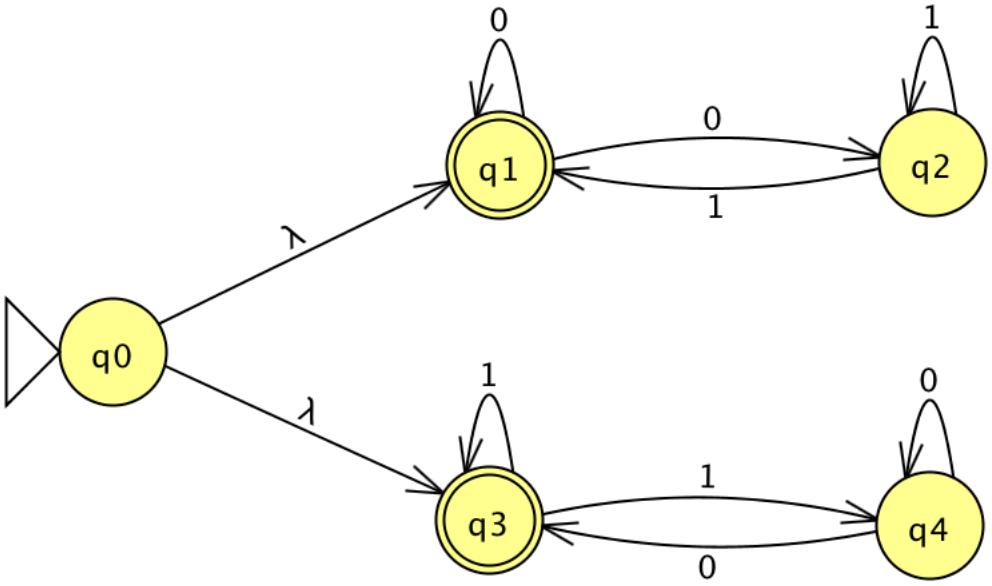
For each of the following languages, decide whether it is regular. If it is regular, give a regular expression or finite automaton that recognizes the language. If it is not regular, demonstrate that using the pumping lemma.

a) Strings containing only the symbol a whose length is a power of 2 (*i.e.* length 2^n)
[The strings `a`, `aa`, `aaaa`, and `aaaaaaaa` are in this language; the string `aaaaa` is not.]

This language is not regular. If it were, it would have a finite automaton capable of recognizing exactly those strings in the language. This automaton would have a number of states – call it p – and the string a^(2^p) would be recognized as in the language by that automaton. However, by the pumping lemma, there is a way to divide this string into a^x a^y a^((2^p) - (x + y)) so that 0<y<=p and a^x a^(n*y) a^((2^p) - (x + y)) is also recognized by this automaton, for all integers n>=0. In particular, a^(x+2y+((2^p)-(x+y)) = a^((2^p)+y) would be recognized by the automaton. But 2^p < (2^p)+y < 2^(p+1) = 2*(2^p) = (2^p) + (2^p), since 0 < y <= p < 2^p. Thus, this string is recognized by the hypothesized automaton, but it is not in the language a^(2^p), since there are no powers of 2 between 2^p and 2^(p+1).

b) All strings with an equal number of occurrences of the substrings `01` and `10`.
[`010` is in this language; `000110` is in the language; `0101010` is in the language; but `010101` is not.]

This language is regular. Here is an accepting automaton:



It is also described by the regular expression (00*11*00*)* | (11*00*11*)*

c) All strings (over {0,1}) consisting of a substring *w* followed by the reverse of the substring.
[The strings `00100100` and `11110101011010101111` are in this language; the strings `00100` and `010101`are not.]

As a preliminary, note that all strings in this language must have an even number of characters.

This language is not regular. If it were, it would have a finite automaton capable of recognizing exactly those strings in the language. This automaton would have a number of states – call it p – and the string 0^p 1 1 0^p would be recognized as in the language by that automaton. However, by the pumping lemma, there is a way to divide this string into 0^x 0^y 0^z 1 1 0^p so that 0<y<=p, x+y+z=p, and 0^x 0^(n*y) 0^z 1 1 0^p is also recognized by this automaton, for all integers n>=0. (Note that x and z may be 0.) For n=0, this means 0^(x+z) 1 1 0^p is recognized by the automaton. But x+z < p, since y > 0. Thus, this string is not a substring followed by the reverse of the substring: The length of this new string is 2p+2-y. Either this number is odd (in which case the string does not meet the precondition that its length is even) or this can be divided into p+1-(y/2) in the substring followed by p+1-(y/2) in the reversed substring. The final p characters of this string are 0s. But p >= p+1-(y/2), since by hypothesis y>0 is even. Thus, all of the characters in the "reversed" substring are 0s; and the two 1s previously at characters p+2 and p+2 in the string are now a part of the initial substring. Since they have no corresponding characters in the "reversed" substring, the second half of the string cannot be a reversed version of the first. Thus, this string is recognized by the hypothesized automaton, but it is not in the language specified.

2. Play the pumping game

Play the pumping game (referenced on the Day 8 page and also found at http://weitz.de/pump/). Solve at least two puzzles from that page (that do NOT appear in question 1, above) and provide the word you chose, the substring the computer chose, and your successfully pumped string.

Notation notes:

- **The notation |w| sub a means the number of a's in the word *w*.**
- ***a^n* means *n* occurrences of a (e.g. *a^8* is aaaaaaaa)**

If you have other questions about notation (or anything else), please post them to Piazza so that we can clarify for everyone.

3. Create a PDA

For one of the non-regular languages in problem 1 or 2 above, create a PDA (preferably in JFLAP) and include it with your completed homework.